

Overview of the Miniature Sensor Technology Integration (MSTI) Spacecraft Attitude Control System

Rob McEwen

Integrated Systems, Inc.

3260 Jay St. Santa Clara, CA 95054

May, 1994

ABSTRACT

Msti2 is a small, 164 kg (362 lb), 3-axis stabilized, low-Earth-orbiting satellite whose mission is missile booster tracking. The spacecraft is actuated by 3 reaction wheels and 12 hot gas thrusters. It carries enough fuel for a projected life of 6 months. The sensor complement consists of a Horizon Sensor, a Sun Sensor, low-rate gyros, and a high rate gyro for despin. The total pointing control error allocation is 6 mRad (.34 Deg), and this is while tracking a target on the Earth's surface. This paper describes the Attitude Control System (ACS) algorithms which include: attitude acquisition (despin, Sun and Earth acquisition), attitude determination, attitude control, and linear stability analysis.

INTRODUCTION

Msti2 is a small, relatively inexpensive spacecraft that was designed, built and tested in 16 months. It is 3-axis stabilized, and will be able to track a point on the Earth's surface to within 6 *m*-rad, (.34°) total pointing error. The control system pointing error allocation (excluding sensor errors) is 1.7 *m*-rad (.1°). The vehicle carries enough fuel for a projected life of 6 months.

Msti2 will be placed in a low-Earth, Sun-synchronous (polar) orbit. The orbital period is 92 minutes. Msti2 will come into contact with a ground station approximately once every 1-2 hours, with a typical contact duration of 8-12 minutes. The infrequent contacts require that the vehicle have a relatively autonomous high-level (mode) controller.

The Msti2 flight code has been designed, coded, and tested in 13 months. This short schedule was, in large part, made possible by the use of automatic code generation. The controls algorithms were first realized in block-diagram form, then automatically converted to Ada code. The generated code comprises approximately 40% of the total flight code. The rest consists mostly of command and telemetry functions, and device specific I/O code, which do not lend themselves readily to block diagram form and hence were hand written. The entire Ada source code is 25,000 lines, not including comments. The executable is slightly less than 128k bytes.

This paper introduces the Msti2 spacecraft from an Attitude Control System (ACS) and software point-of-view. The paper then describes various aspects of the ACS design, such as attitude determination, attitude control, linear analysis and design, and robustness.

PHYSICAL CHARACTERISTICS, SENSORS AND ACTUATORS

The vehicle is octagonal in shape, and weighs 164 kg (362 lb) including fuel and payload. The principle moments of inertia are (17, 20 14) kg-m². It is approximately .8 m (2.5 ft) in diameter, and 1.4 (4.5 ft) in height. The average power consumption is 145 watts.

The CPU is a 1750A with 1 M-byte of sRAM, and 512 k-bytes of E²PROM. It can process approximately .8 MIPS. The CPU communicates with the interface boards through a VME bus.

Three reaction wheels provide actuation for fine pointing control. The wheel motor torque capacity is at 25 milli-Newton-meters. The drag torque ranges from 2 to 15 *m*-Nm. The wheel can spin up to 530 rad/sec (5100 rpm), which corresponds to a momentum storage capacity of 4 Newton-meters-seconds. A digital tachometer measures wheel speed at 54 counts per revolution.

The vehicle carries 12 monopropellant hydrazine thrusters for wheel desaturation, orbit adjust, and backup attitude control. There are 8 ACS thrusters at .9 Newtons (.2 lb), and 4 Delta-v thrusters at 4 Newtons (1 lb).

The Sun sensor has a 128° field of view in each of two axes, with a quantization step of 34 μ -rad per lsb. Static error is about 26 lsb, 3σ . The sun sensor faces out along the +*y* vehicle axis.

The Horizon sensor has a conical scan of 60° at 5 Hz, with a blanking region of 81°. Quantization is 96 μ -rad/lsb. Sensor error is 154 lsb, 3σ . The scan cone faces out along the -*y* vehicle axis.

The high resolution gyros are rate-integrating, with a quantization step of 20 μ -rad/lsb. Noise is approximately .7 lsb, 3σ . The saturation level is $\pm 5^\circ$ /sec (87 μ -rad/sec).

DESCRIPTION OF THE MSTI2 ATTITUDE CONTROL SYSTEM

Overview

This section introduces the major components of the Msti2 Attitude Control System (ACS). Each component is discussed in further detail in subsequent sections.

Figure 1 shows an overview of the Msti2 ACS. Each block represents a distinct part of the control system. All blocks represent software, except the Watchdog Timer, and the sensors.

The architecture in Figure 1 is due to the basic control requirements, which are:

1. Despin the vehicle after separation from the booster.
2. Point the +*y* vehicle axis at the Sun to illuminate the solar array, and to obtain a Sun sensor measurement.
3. Point the +*z* axis of the vehicle at nadir to obtain a Horizon Sensor measurement, and to establish communication with the primary antenna.
4. Perform an orbit adjust maneuver.
5. Track a point on the Earth's surface to within 6 *m*-rad total pointing error, and 100 μ -rad/sec rate stability.

The block "Modes" contains algorithms that accept signals from the sensors, and compute position and rate error. The purpose of the first 6 modes is to despin the vehicle, after separation from the Scout booster, and then locate the Sun and the Earth. This process takes about 10 minutes.

All modes output position and rate error in the form of a normalized quaternion, and an angular velocity vector.

Only one mode may be active at a time. Mode selection is based on boolean variables such as: body rates small, Sun present, Earth present, etc. The block "Mode Control Logic" accepts

inputs from the sensors and activates the appropriate mode. Automatic mode selection may be overridden by ground controllers at any time by either time-tagged or immediate commands. The Modes and Mode Controller are described further in the following section.

The remaining four Mission modes control precision pointing and maneuvering during payload operation. These modes differ from the others in that they have a complete, large-angle attitude reference, and will maintain attitude reference should the Sun or the Earth or both go out of the field of view during a maneuver.

The "Command Generator" block computes the desired attitude and body rates for each of the four Mission modes. The output of this block is an inertial quaternion and inertial body rates.

The "Attitude Controller" block takes the position and rate error computed by one of the modes, and requests a torque. This block performs a proportional derivative (PD) computation when the angular position error is small enough. Otherwise, it does a constant rate slew about the Euler axis until the error is small enough to switch to PD.

The requested torque vector is routed to either the wheel controller, or the thruster controller. Either, or both, may be enabled.

The "Thruster Controller" converts the requested torque vector into on-times for each of the 12 thrusters, using pulse width modulation. Any subset of the 12 thrusters may be enabled by ground commands. This controller also contains logic to maintain attitude control during orbit adjust by off-modulating the 4 delta-v thrusters.

The "Wheel Controller" takes the requested torque from the Attitude Controller, and computes wheel motor commands. This controller contains a speed feedback loop based on the wheel tachometer. The wheel controller also will accept speed set-point commands from the ground for desaturation.

Modes and Mode Controller

Figure 2 shows the Mode Controller, which is a set of recursive boolean equations in state diagram form. This is an actual block diagram, from which Ada code was generated, and not a cartoon representation.

Each of the small rectangular boxes along the transition path represent the conditions which must be satisfied for transition.

The mode transition conditions are summarized in Table 1. Whenever a comparison of a sensor value to a commandable threshold is made, it is simply stated as "big" or "small."

Two important mode controller commands are MODELIM and MODEFORC. MODELIM sets an upper limit beyond which the mode controller cannot go. For example, MODELIM is initially set to 6 for orbit injection, because the orbit propagator must first be manually initialized by the ground controllers before any of the Mission modes can have a valid attitude reference. For this reason, once the vehicle gets to Mode 6 it must wait for ground commands at a subsequent pass.

MODEFORC overrides the transition conditions and forces the mode controller to enable a commanded mode.

These two commands are not shown Table 1 because they apply to every mode. Keep in mind, for example, that mode 1 is reached by setting MODELIM to 1.

There is safemode logic which can override the Mode Controller at any time. The logic checks the following 6 conditions against commandable thresholds:

1. Sun absent too long

Transition	Up	Down
1 ↔ 2	None.	None.
2 ↔ 3	Body rates small.	Body rates large.
3 ↔ 4	Sun Present.	Sun not present or Body rates large.
4 ↔ 5	Body rates small and Sun pointing error small.	Sun not present or Sun pointing error large or Body rates large.
5 ↔ 6	Earth present and Pitch error small.	Sun not present or Earth not present or Body rates large.
> 6	Body rates small.	None.

Table 1: Mode Transition Conditions

2. Earth absent too long
3. Body rates too large
4. Pointing error too great
5. Wheel speed too high
6. Thrusters on too long

Should any of these conditions be met, a corresponding safemode delay queue will execute. These queues contain sequences of preset commands. Additionally, ground controllers can reprogram these queues, so that the vehicle can be set up to react properly should there be any specifically anticipated anomalies.

Each of the modes is described below. Recall that only one mode may be active at a time, and that each mode puts out position and rate error in the form of an error quaternion and an error angular velocity.

The attitude determination algorithm is enabled for all modes, that is, it is always running, although the attitude it computes is ignored for Mode 6 or less.

Mode1: Idle. The requested torque output of the attitude controller is set to zero. The wheel controller will maintain wheel speeds. No sensors are used for feedback.

Mode2: Capture. The requested torque is proportional to the body rates as measured by the despin gyro. Nominally, the actuators are thrusters 1-4, but this can be reset to any combination of thrusters by ground commands. This mode will damp nutation for the expected range of nutation frequency.

Mode3: Sun Search. The mode controller activates this mode when the measured body rates drop below a commanded threshold. This mode propagates gyro rates to form a large angle quaternion. The initial reference frame is inertially fixed, and is taken to be wherever the vehicle is when the mode is enabled. Relative to this initial frame, the vehicle performs a series of 3 slews designed to pass the entire sky through the field-of-view of the Sun sensor. The sequence is a -270° yaw, followed by a 90° roll, followed by a 180° yaw.

As soon as the Sun present flag is set, the mode controller transitions to Sun point. If the Sun is not found, the vehicle does a final maneuver to return to the initial orientation, and sets a "Sun Not Found" flag in telemetry.

The mode uses only gyros for input.

Mode4: Sun Point. The vehicle will slew its $+y$ axis, where the solar array is, into coincidence with the Sun-line. The Sun sensor outputs are then passed through for roll and yaw position control. Pitch angle is derived by essentially accumulating the pitch gyro channel. A command will set a pitch rotation rate.

The Sun pointing mode uses both the gyro and the Sun sensor for input. When both the body rates and the Sun pointing error are below commandable thresholds, the mode controller transitions to Earth Search.

Mode5: Earth Search. This is basically the same as the previous mode, except that the pitch rate is increased, and the mode controller will activate Sun/Earth pointing mode as soon as the Earth is far enough within the field-of-view of the Horizon sensor.

Mode6: Sun/Earth Point. Here, the pitch angle is now computed from the horizon sensor. Each vehicle axis now has a measured position and rate term. All three attitude determination sensors are in use.

This mode forms the three measured position signals into an approximate small angle quaternion. The quaternion represents the angular displacement of the vehicle from a reference frame which has its y axis centered on the Sun, and z axis as close to the Earth as possible. The z axis of this frame will vary seasonally from approximately 7° to 30° off Earth nadir due to orbit geometry and the inclination of the Earth to the Ecliptic.

The following "Mission" modes use an exact, large angle quaternion computed by the attitude determination. This computation requires on-board propagation of the Sun vector and the vehicle orbit state. After orbit injection, the orbit propagator must be initialized with a measured ephemeris before the attitude determination is valid. Once this is accomplished, any of the following modes may be selected by ground commands.

The attitude determination propagates gyro outputs in order to compute position when the Sun or the Earth or both are out of the field-of-view of their respective sensors.

The Mission modes have the following function:

Mode7: Mission-Local Vertical Hold. The vehicle is held fixed with respect to the Local-Vertical Local-Horizontal reference frame. Ground controllers may command the vehicle to a constant offset from this frame, while in this mode.

Mode8: Mission-Slew. In addition to the automatic slew submode of the attitude control, ground controllers may explicitly command a slew. The slew acceleration, rate, total slew angle, and slew axis are all commandable. The slew profile is a constant acceleration, a constant rate, and a constant deceleration about an inertially fixed Euler axis.

Once the slew is completed, the vehicle maintains the final inertial orientation and awaits ground commands.

Mode9: Mission-Surface Point Track. For the payload to acquire its target, its line-of-sight must remain on a predetermined point on the surface of the Earth. This mode takes target latitude, longitude and altitude as input. It then maneuvers the vehicle so that the z body axis points at the target. It also keeps the vehicle yz plane centered on the Sun to maintain solar array power.

Mode10: Mission-Inertial Hold. The vehicle is held fixed with respect to the ECI frame. The flexibility and capability of these modes is greatly expanded by use of the delay queues.

Delay queues are buffers in which ground controllers may store a series of time-tagged commands. The flight code will then automatically execute these commands at the prespecified times. Time-tagged commands may then enable or disable modes, or reset parameters. For example, LV hold, Mode 7, can be given a series of offsets, at regular intervals, so that the vehicle in effect can be made to track a prespecified orientation profile.

CONTROL DESIGN

Attitude Determination and Command Generation

The attitude determination involves four frames of reference, shown in Figure 3.

The first is the Earth Centered Inertial (ECI) frame, which has its x - y vectors in the equatorial plane, and the x vector points at the Sun on the vernal equinox. The on-board orbit and Sun propagators output vectors coordinatized in these frames.

The local vertical (LV) frame has the z axis pointing to the center of the Earth, and the x axis in the direction of flight.

The Mission frame represents the desired vehicle orientation, and may move with respect to the local vertical frame. For example, during payload operation, the z axis of the Mission frame will track a point on the Earth's surface. In the Local Vertical Hold mode, the Mission frame is coincident with the local vertical frame.

The fourth frame is fixed in the vehicle body. The objective of the attitude control is to minimize the angular error between the Mission frame and the body frame.

A quaternion with subscript notation specifies the orientation of one frame with respect to another. For example, the orientation of the Body frame with respect to the Mission frame is denoted $q_{B/M}$. The angular velocity of the Body frame with respect to the Mission frame is $M\omega^B$. This nomenclature is used throughout this paper.

The attitude determination propagates gyro outputs in order to compute position when the Sun or the Earth or both are out of the field-of-view of their respective sensors.

In typical control design procedure, one would linearize the plant, and then implement a linear state estimator for attitude determination. For Msti2, payload operation requires that the vehicle z axis track a point on the Earth's surface, which in turn requires that the vehicle have angular acceleration and deceleration over large angles. This makes linearization of Euler's equation infeasible. For this reason, the algebraic method, as described in Wertz, page 424, was chosen for attitude determination. This method provides an exact large angle quaternion over a wide range of vehicle orientations, and does not require a linearized model of the vehicle dynamics.

The attitude determination algorithm passes the algebraically computed quaternion through a low-pass filter to reduce quantization noise.

The algorithm produces $q_{B/ECI}$, the quaternion expressing body attitude with respect to the Earth Centered Inertial (ECI) reference frame. This computation requires orbital location and Sun position (coordinatized in ECI) as input. For this reason, the flight software contains both Sun and vehicle orbit propagation algorithms.

The orbit propagation algorithm must be initialized with a measured ephemeris after orbit injection. Ground operators must subsequently perform periodic ephemeris updates to correct propagator drift. The propagator contains an aerodynamic drag term, which has a commandable value that will be used to adjust the propagator to minimize drift.

The command generator computes desired vehicle orientation and rates with respect to the ECI frame. This orientation is expressed as $q_{M/ECI}$, where M symbolizes the "mission"

(or desired) reference frame. The attitude error is then

$$q_{B/M} = Q_{B/ECI} q_{ECI/M} \quad (1)$$

This quaternion multiply is expressed here as a matrix multiply where Q is the 4 by 4 matrix representation, and $q_{ECI/M}$ is the conjugate of $q_{M/ECI}$. $q_{B/M}$ is passed to the attitude control compensation, and represents pointing error.

The command generator can generate 4 different orientation and rate profiles for the mission frame, which correspond to the 4 Mission Modes described previously.

Attitude Control

The attitude control algorithm, as shown in Figure 4, computes a requested torque T_r , based on position and rate errors. The basic scheme is proportional-derivative, with slightly different variations for despin, large angular error, and small angular error.

During despin, requested torque is based on angular velocity:

$$\vec{T}_r = -I^B 2K_r {}^{ECI}\vec{\omega}^B \quad (2)$$

where I^B is the vehicle inertia matrix, K_r is the rate gain, and ${}^{ECI}\vec{\omega}^B$ is the angular velocity of the body (vehicle) with respect to the ECI frame.

Nutation motion is damped by this control because the applied torque is always proportional to the negative of the angular velocity. However, the control system sampling rate must be significantly faster than the nutation frequency or else the sampling process will cause a potentially destabilizing delay.

The nutation frequency ω_N , is defined here as the frequency at which the angular velocity vector rotates around the principal axis *as viewed by an observer fixed in the body frame*. Then,

$$\omega_N = \left| \frac{I_s}{I_t} - 1 \right| \omega_{sp} \quad (3)$$

where ω_{sp} is the spin frequency, and I_s , I_t are the spin and tangential inertias, respectively. For Msti2, $\omega_N \approx 4$, or about .6 Hz, which is sufficiently less than the control system sampling frequency of 5 Hz.

When the control is passed a large angle $q_{B/M}$, the requested torque is

$$\vec{T}_r = I^B \left(\omega_s \frac{\vec{q}}{\|\vec{q}\|} - {}^{ECI}\vec{\omega}^B \right) K_r + {}^{ECI}\vec{\omega}^B \times I^W {}^B\vec{\omega}^W \quad (4)$$

where \vec{q} is the vector part of $q_{B/M}$, I^W is the inertia matrix of the three reaction wheels, and ${}^B\vec{\omega}^W$ is the angular velocity of the wheels with respect to the body frame. ω_s is a (scalar) commanded slew rate.

This causes a rate controlled slew about the Euler axis. The second term above accounts for stored wheel momentum.

When $q_{B/M}$ represents a small angular error, the requested torque is

$$\vec{T}_r = -I^B (2K_p \vec{q}_{B/M} + K_r {}^M\vec{\omega}^B) + {}^{ECI}\vec{\omega}^B \times I^W {}^B\vec{\omega}^W \quad (5)$$

This essentially is PD control. K_p and K_r are position and rate gains, and ${}^M\vec{\omega}^B$ is the angular velocity of the body (vehicle) with respect to the mission frame.

Linear Modelling and Stability Analysis

Figure 5 shows the small-signal linear model of the attitude control loop. The following analysis is for the fine pointing case, which uses wheel control. Thruster actuated attitude control is not discussed here.

The outer loop two loops represent the small signal behavior of the attitude control. The inner loop, from \bar{T}_{in} to the zero order hold, is the wheel controller.

The Wheel Controller takes the requested torque from the Attitude Controller, and computes wheel torquer motor commands.

The wheel controller contains a feedforward path that passes the requested torque to the wheel motors, after a conversion to counts. A feedback term, based on the tachometer, is added to the motor command. The primary purpose of the feedback loop is to compensate for wheel bearing drag, which is a nonlinear function of wheel speed, and is also temperature dependent.

The feedback term is computed by first scaling the requested torque by the wheel inertia matrix, and then accumulating (integrating) it to produce a wheel speed command. The filtered tachometer output is differenced with the speed command to produce an error signal, which is passed through a compensator.

The compensator contains a proportional term and an integral (accumulated) term. The purpose of the accumulator is to command just enough torque to cancel the drag torque.

The accumulator state is limited because the controller may ask for more torque than the wheel motor can provide (this is nonlinear behavior). For example, this will happen during wheel desaturation when the wheel speed set point is reset to zero. During the time the wheel is spinning down, the accumulator, if not limited, would be "winding up," or accruing a large state, which would cause a large, slow, and undesirable transient.

The tachometer measurement is passed through a first order filter to reduce quantization noise.

The initial ACS design iteration was done by a series of simplifying approximations. The quantization filters have a bandwidth of .5 Hz, which is much higher than the expected closed loop bandwidth, so they may be neglected for approximate analysis. The torque control (T_o/T_i) has an approximately unity transfer function so it may be neglected as well. The closed loop bandwidth is much lower than the sampling frequency, so the whole system may be approximated by its continuous-time equivalent. These approximations result in the standard second order continuous-time transfer function shown below:

$$\frac{\theta_C}{\theta_R} = \frac{K_p}{s^2 + K_r s + K_p} \quad (6)$$

This is a type 1 system, and has no steady-state error for step inputs.

The specifications for control system performance are 1.7 m-rad pointing error and 100 μ rad/sec rate stability during payload operation. The vehicle will probably be maneuvering during payload operation, and so the natural frequencies of the control loop should be much higher than the frequency content of the commanded angles and rates. This indicates that the dominant time constants should be less than about 5 seconds.

As an initial value, the time constant is chosen to be 2.5 sec, which implies that $K_r = .8 \text{ sec}^{-1}$. It is desirable to keep

$$\frac{1}{\sqrt{2}} \leq \zeta \leq 1 \quad (7)$$

which implies that that

$$.16 \leq K_p \leq .32 \quad (8)$$

$$.04 \leq f_{BW} \leq .09 \quad (9)$$

where f_{BW} is the closed loop bandwidth in Hz. The velocity error constant indicates that pointing performance will improve as K_p increases, however, stability margins decrease and sensor noise amplification increases.

The loop transfer function, with the loop opened at the torque command input to the plant (K_p gain in Figure 5) is

$$l(s) = \frac{K_r s + K_p}{s^2} \quad (10)$$

This system has an infinite gain margin, and a phase margin between 65° and 75° . Of course, these values discussed above are approximate, and the stability margins in particular are suspect, because the loop must be opened at the zero-order-hold in Figure 5, instead of at K_p , to get a realistic measure. Accurate values can be obtained by a numerical analysis of the system in Figure 5.

Starting with the (K_p, K_r) values above, the system in Figure 5 was numerically discretized and analyzed. Then several trial-and-error simulation runs were made, involving both the linear analysis program, and the nonlinear truth simulation, to slightly adjust the gains to achieve the desired performance and robustness. The final results are summarized below.

Kp = 2.5000D-01 1/sec**2, Position Gain
 Kr = 9.0000D-01 1/sec, Rate Gain
 fquant = 5.0000D-01 Hz, Quantization Filter Bandwidth
 Krw = 6.0000D-02 1/sec, Wheel Rate Gain
 Kiw = 1.0000D-03 1/sec**2, Wheel Integrator Gain
 fwhl = 1.0000D-01 Hz, Wheel Tach Filter Bandwidth
 The drag torque coefficient (viscous) is 4.5000D-05 Nm/rad/sec.

The order of the open loop system is: 7
 The open loop roots are:

---Z Root Information Algorithm-----
 1 eigenvalues at z=0 have been removed.
 There is (are) 6 real evals, 0 complex pairs
 and 0 s=0 evals.

Magnitude and Phase (Deg):
 0.5335 0.0000 (Quantization filter pole)
 0.8819 0.0000 (Wheel tach filter pole)
 0.9988 0.0000 (Drag Torque pole)
 1.0000 0.0000
 1.0000 0.0000
 1.0000 0.0000

Time constant, 3 time constants (seconds):
 0.3183 0.9549
 1.5915 4.7746
 171.1135 513.3406 (Wheel drag time constant)

---End Z Root Information Algorithm-----

The gain crossover frequency is 1.5180D-01 Hz.
The gain margin is: -2.4418D+01 1.3529D+01 dB.
The phase margin is: 4.1615D+01degrees.

The order of the closed loop system is: 7
The closed loop roots are:

---Z Root Information Algorithm-----
1 eigenvalues at z=0 have been removed.
There is (are) 2 real evals, 2 complex pairs
and 0 s=0 evals.

Magnitude and Phase (Deg):
0.8165 0.0000
0.8501 9.0693
0.8501 -9.0693
0.9307 0.0000
0.9940 0.1137
0.9940 -0.1137

Damping ratio, Natural Frequency (Hz), time constant, 3*T.C.:
0.7161 0.1805 1.2315 3.6944
0.9493 0.0050 33.3692 100.1075

Time constant, 3 time constants (seconds):
0.9868 2.9604
2.7852 8.3555

---End Z Root Information Algorithm-----

The closed loop bandwidth is 6.8405D-02 Hz.

The wheel accumulator time constant shows up here as 33 seconds. From a transfer function point of view, these poles are nearly cancelled by zeros, so the effect on the input-output response is small. The wheel accumulator state is limited to a commandable value between 4-15 m-Nm, so this transient will not show up anyway.

Simulations show that the step response is dominated by the 2.9 second time constant.

Robustness

This discussion pertains to the fine pointing wheel control loop in Figure 5. Thruster control is not analyzed here.

The above linear analysis showed a gain margin of 13 db, and a phase margin of 47°, with the loop opened at the wheel motor interface. This point is inside both the attitude control and the wheel control loops. It is important to measure the stability margin at this point since it is where the uncertainty in the plant is best represented. The uncertainty is caused by the torquer motor coefficient, vehicle and wheel inertias, etc.

The linear analysis also shows a lower gain margin of 24 dB. This is due to the accumulator pole which produces the commanded wheel speed setpoint. It is acceptable because if the gain at that point should drop by 24 dB, the wheel control would be so weak that the vehicle would have to be switched to thruster control anyway.

Another point of interest is the drag torque, since it is not constant, and may vary by a factor of about 7. It is known to be a function of both wheel speed and temperature. The stability margin at the drag torque coefficient, with all other loops closed, is 66 dB gain

margin, and infinite phase margin. This shows that the system is stable for all realistic drag torque values.

ACKNOWLEDGEMENTS

I would like to acknowledge the other members of the Msti2 ACS and Software design team, who have contributed significantly to this project. They are: Fred Tubb, Keiko Chevray, Andy Waddell, John Farazian, and Chris Rygaard.

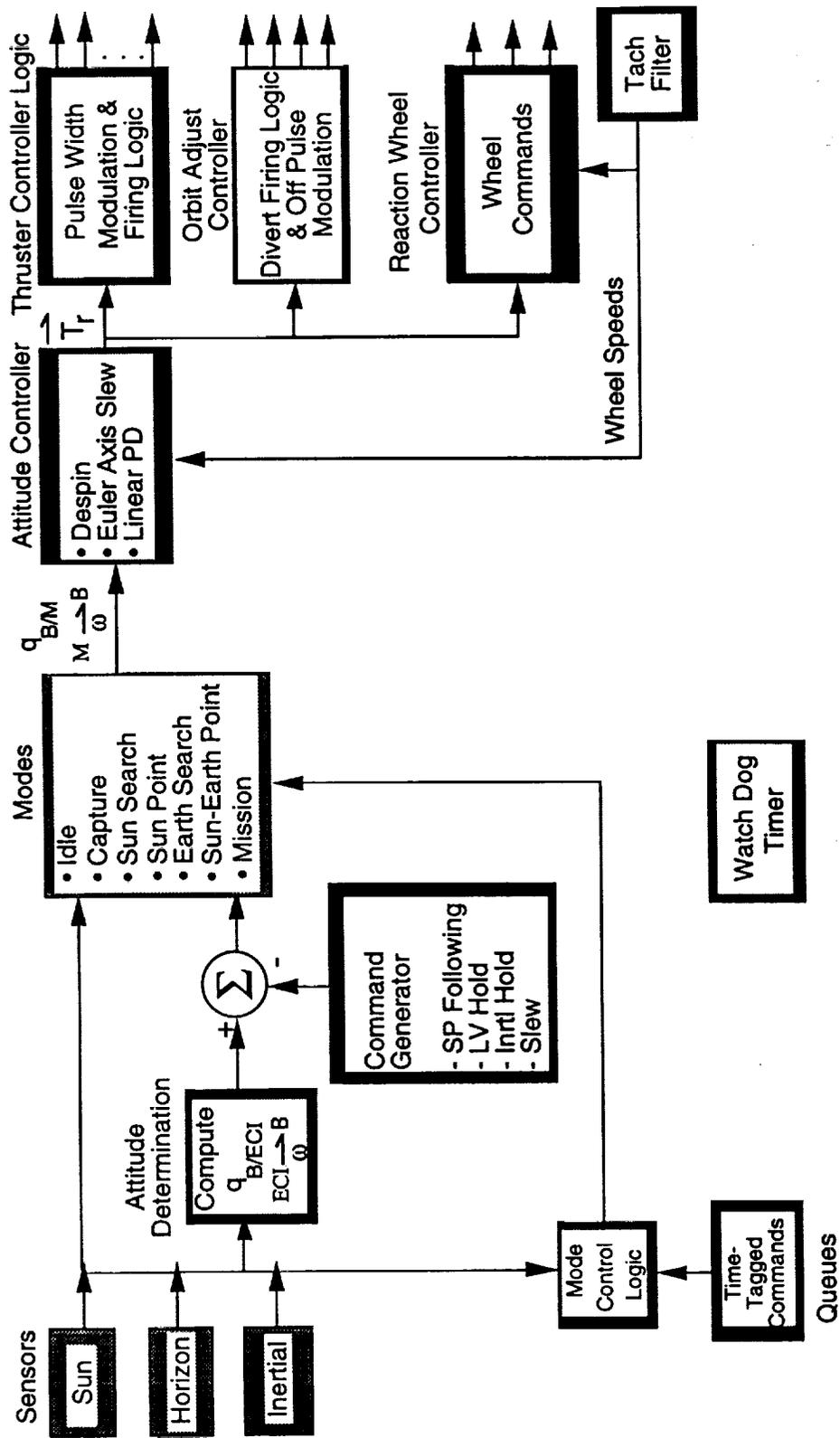


Figure 1: Overview of Msti2 ACS

State Diagram	STD.Inputs	STD.Outputs	SuperBubble	Level
Mode Controller	10	10	TOP_LEVEL	1

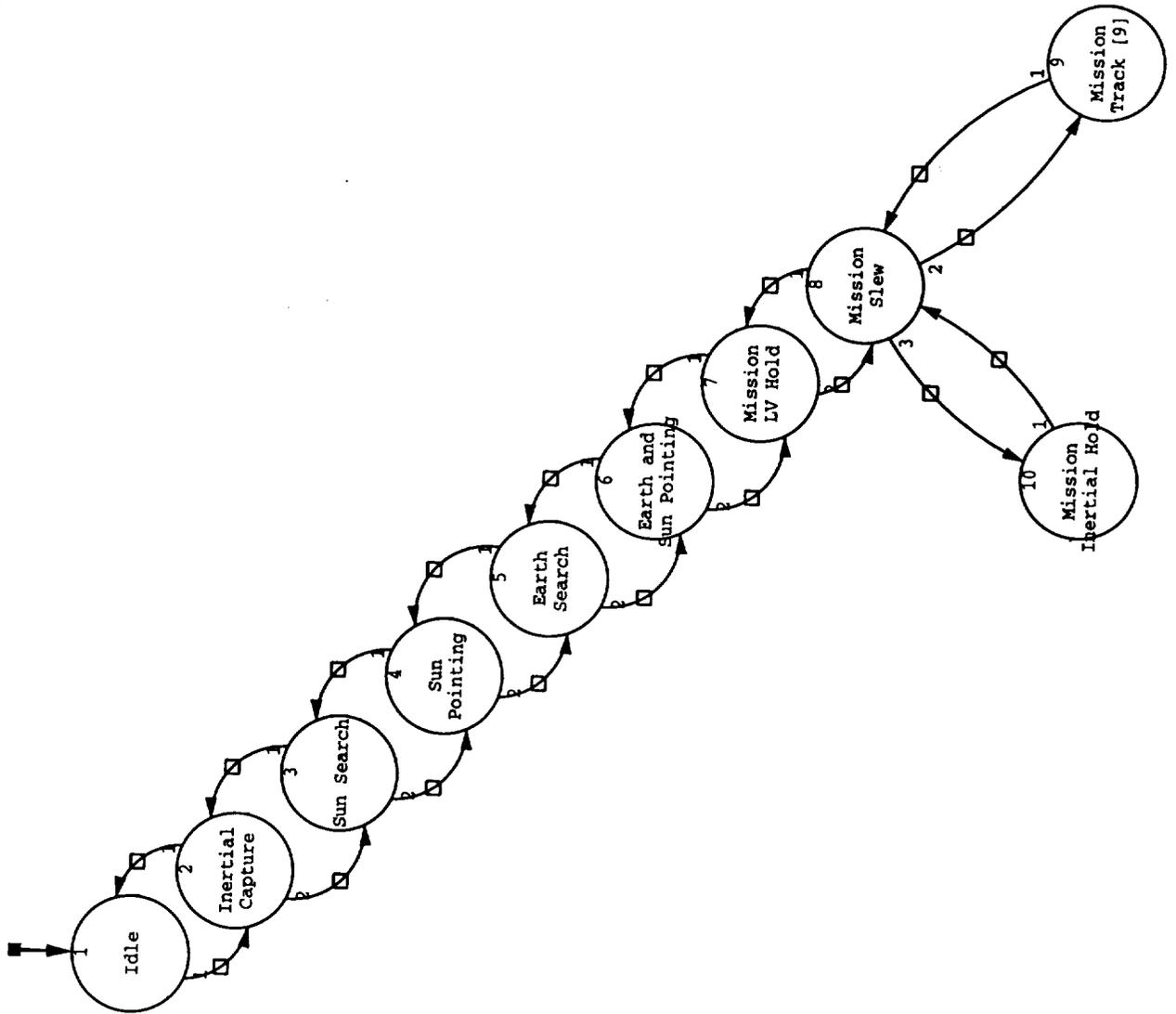


Figure 2: Mode Controller State Diagram

